



# A generic off-line approach for dealing with uncertainty in production systems optimisation

Alexis Aubry, André Rossi, Mireille Jacomino

## ► To cite this version:

Alexis Aubry, André Rossi, Mireille Jacomino. A generic off-line approach for dealing with uncertainty in production systems optimisation. 13th IFAC Symposium on Information Control Problems in Manufacturing, INCOM'2009, Jun 2009, Moscou, Russia. pp.1464-1469, 10.3182/20090603-3-RU-2001.00244 . hal-00393745

**HAL Id: hal-00393745**

**<https://hal.science/hal-00393745>**

Submitted on 9 Jun 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A generic off-line approach for dealing with uncertainty in production systems optimisation

Alexis Aubry\* André Rossi\*\* Mireille Jacomino\*\*\*

\* CRAN, Nancy-Université, CNRS, Boulevard des Aiguillettes  
B.P. 70239 F-54506 Vandœuvre lès Nancy  
(e-mail: alexis.aubry@cran.uhp-nancy.fr).

\*\* Lab-STICC, Université de Bretagne-Sud, Centre de Recherche  
F-56321 Lorient  
(e-mail: andre.rossi@univ-ubs.fr)

\*\*\* G-SCOP, INPGrenoble, UJF, CNRS, 46 avenue Félix Viallet  
F-38031 Grenoble Cedex  
(e-mail: mireille.jacomino@inpg.fr)

---

**Abstract:** This paper addresses production systems optimisation in uncertain context. A standard framework for solving such type of problem is depicted in a 3-step approach. The two first steps consist of off-line characterisation of the problem and calculation of solutions with some desired performance. A generic approach to implement these off-line steps is introduced in this paper. This approach relies on calculation of robust off-line solutions. First a generic framework of robustness is defined. Then five standard optimisation problems are derived and related to the so called stability and sensitivity analysis. This generic approach is then applied to a multi-purpose machines problem.

*Keywords:* Production systems, Optimisation, Uncertainty, Robustness.

---

## 1. INTRODUCTION

For a long time, works made in optimisation field assumed that optimisation problem data were known and sure. The goal of the optimisation exact methods was to build an optimal solution for a given instance that was supposed to represent the state of the real system (modelled by the optimisation problem).

However, optimisation problems are only imperfect models which do not always fit reality. Model accuracy is often put in question because of input data deviation. In such a context, putting a so called “optimal solution” in practice (calculated for a forecast instance) can lead to poor performance. The solution may even no longer be admissible because some constraints may be violated. This is why the general problem of model parametrisation is particularly relevant for combinatorial optimisation. Indeed such problem aims at finding the best solution among a set of unrelated solutions.

Taking uncertainties into accounts in optimisation is a real challenge and a lot of works can give this evidence (see (Aubry, 2007) for a commented survey of robustness approaches). Developing robustness features has appeared to be an efficient way to cope with uncertainties (Kouvelis and Yu, 1997; Vincke, 1999; Perny and Spanjaard, 2006) even though researchers do not agree on its definition. Roughly speaking, robustness measures the solution ability to remain “good” despite uncertainties. What is exactly a so called good solution and the considered class

of uncertainties is strongly application-dependent, and has led researchers to develop a large variety of approaches. The goal of this paper is to present a generic off-line approach to deal with uncertainty in optimisation. In section 2, the usual approach for dealing with optimisation problem when data numerical values are supposed to be known is recalled, and its limits are highlighted if the approach is to be applied in uncertain context. A Three steps resolution process of optimisation problems under uncertainty is shown in section 3. In section 4, a generic off-line approach is proposed. A robustness definition is given and five robustness issues, which are highlighted by the robustness definition, are detailed, discussed, compared with existing literature and finally illustrated through an example in section 5.

## 2. THE LIMITS OF OPTIMISATION IN CERTAIN CONTEXT

Classically, solving an optimisation problem  $\mathcal{P}$  consists in building a solution  $\mathcal{S}$  that optimises a criterion  $z$  (that will be considered as a minimisation criterion without loss of generality) and satisfies some constraints provided that the problem data are known and sure. A forecast instance  $\mathcal{I}_{ref}$  is used to compute  $\mathcal{S}$  and the obtained optimal solution is applied to this instance. Getting an optimal solution is yet a problem that is often hard.

An optimal solution to the problem  $\mathcal{P}$  for an instance  $\mathcal{I}$  is denoted  $\mathcal{S}_{\mathcal{I}}^*$  and its performance is denoted  $z_{\mathcal{I}}^*$ . The

performance of a solution  $\mathcal{S}$  applied to an instance  $\mathcal{I}$  relatively to an optimisation criterion  $z$  is denoted  $z_{\mathcal{I}}(\mathcal{S})$ .

The classic way to solve an optimisation problem without uncertainty is the predictive approach. An off-line algorithm builds an optimal solution  $\mathcal{S}_{\mathcal{I}_{ref}}^*$  for the forecast instance  $\mathcal{I}_{ref}$ , and guarantees an optimal performance for this instance only, valued by  $z_{\mathcal{I}_{ref}}^*$ . In practice, the real system is subject to perturbations such that the solution  $\mathcal{S}_{\mathcal{I}_{ref}}^*$  is applied to the actual instance  $\mathcal{I}$  that may be different from the forecast instance  $\mathcal{I}_{ref}$ , and  $\mathcal{S}_{\mathcal{I}_{ref}}^*$  may even not be admissible. In the most optimistic scenario (when the solution remains admissible for  $\mathcal{I}$ ), the actual performance  $z_{\mathcal{I}}(\mathcal{S}_{\mathcal{I}_{ref}}^*)$  can be “far” from the forecast performance  $z_{\mathcal{I}_{ref}}^*$ , leading a costly resolution step to return a poor actual performance solution.

The limits of the classic approach are clearly appearing: it does not ensure any performance for the actual problem instance. Consequently, taking the uncertain context into account is necessary for addressing optimisation problems that model real-life production systems.

In the following of the paper, uncertainty and perturbations are considered as synonyms.

### 3. A FRAMEWORK FOR DEALING WITH UNCERTAIN CONTEXT

In (Billaut et al., 2005), the authors propose a complete framework for addressing scheduling problems under uncertainty that can be generalised to optimisation under uncertainty. This framework is split into three steps:

(1) Step 0: static problem definition

Classic specifications of the optimisation problem without considering perturbations are given with the optimisation criterion  $z$ . Moreover, perturbations are specified with their model ( $\mathcal{P}$  is no more defined by just one instance but by a risk to be covered: a set of instances in this paper). The notion of performance of a solution must be defined by answering the question: “what must be guaranteed despite perturbations?”

(2) Step 1: calculation of a set of solutions by an off-line algorithm

The off-line algorithm uses available knowledge about perturbations (the risk to be covered defined in step 0), to build a set of robust solutions guarantying a performance on the considered instances (as defined in step 0). Instead of computing an optimal solution for a single forecast instance, a global performance on a set of instances is preferred. Such solutions if there exist, are said to be robust on the considered instances.

(3) Step 2: calculation of a single solution by an on-line algorithm

The on-line algorithm uses the progressive knowledge about reality to build a single solution. This solution is either chosen among the set of solutions calculated in step 1 or is the result of a reparation of the current solution or the calculation of a new solution because an unexpected scenario has occurred.

The approach that is presented in this paper covers steps 0 and 1.

### 4. A GENERIC OFF-LINE APPROACH TO DEAL WITH PERTURBATIONS

In this section, we present a generic off-line approach to solve an optimisation problem under uncertainty by splitting it into several relevant robustness issues. This approach is based on the following robustness definition.

#### 4.1 Robustness of a solution

Many robustness definitions can be found in the literature as shown in (Aubry, 2007). The robustness definition given in (Rossi, 2003) generalises many other ones and is used in this paper: robustness is defined as the ability for a solution to guarantee a performance level  $\mathcal{L}_\lambda$ , according to a robustness criterion  $\lambda$ , for each instance belonging to a risk to be covered  $\mathcal{P}$ . The robustness criterion  $\lambda$  evaluates a global performance on the set  $\mathcal{P}$ . The usual robustness criteria have been defined in (Kouvelis and Yu, 1997) as follows:

- Absolute robustness:  $\lambda_1(\mathcal{S}, z, \mathcal{P}) = \max_{\mathcal{I} \in \mathcal{P}} z_{\mathcal{I}}(\mathcal{S})$
- Robust deviation:  $\lambda_2(\mathcal{S}, z, \mathcal{P}) = \max_{\mathcal{I} \in \mathcal{P}} (z_{\mathcal{I}}(\mathcal{S}) - z_{\mathcal{I}}^*)$
- Relative robustness:  $\lambda_3(\mathcal{S}, z, \mathcal{P}) = \max_{\mathcal{I} \in \mathcal{P}} \frac{(z_{\mathcal{I}}(\mathcal{S}) - z_{\mathcal{I}}^*)}{z_{\mathcal{I}}^*}$

It is recalled that one purpose of a robustness criterion is to assess global performance (i.e. on a set of instances).

*Robustness mathematical formulation.*

*Definition 1.* In (Rossi, 2003), a solution  $\mathcal{S}$  is said to be  $\mathcal{L}_\lambda$ -robust on the set of instances  $\mathcal{P}$  relatively to the robustness criterion  $\lambda$  if it satisfies the following inequality:

$$\lambda(\mathcal{S}, z, \mathcal{P}) \leq \mathcal{L}_\lambda \quad (1)$$

*$\mathcal{L}_\lambda$ -robustness and step 0.* Step 0 in the resolution framework is completed by stating the optimisation problem under uncertainty using the  $\mathcal{L}_\lambda$ -robustness. In fact, defining  $z$  is specifying the optimisation criterion, and defining  $\mathcal{P}$  is modelling the perturbations that should be taken into account. Choosing  $\lambda$  and setting  $\mathcal{L}_\lambda$  is answering the question: “what must be guaranteed despite perturbations?” as  $\lambda$  evaluates the global performance of  $\mathcal{S}$  on the set of instances  $\mathcal{P}$  and  $\mathcal{L}_\lambda$  gives the level of performance to be guaranteed by the solution on  $\mathcal{P}$  (as a risk to be covered).

#### 4.2 Some robustness issues

The  $\mathcal{L}_\lambda$ -robustness highlights five robustness issues. These issues are identified by their input data or decisions variables in the definition of  $\mathcal{L}_\lambda$ -robustness. In the following of the paper  $\lambda$  and  $z$  are assumed to be given.

*First robustness issue: stability analysis.* It is assumed that in step 0, only a forecast instance  $\mathcal{I}_{ref}$ , a performance level  $\mathcal{L}_\lambda$  and a solution  $\mathcal{S}$  can be defined to characterise the optimisation problem. Then, the first robustness issue can be stated as follows: “knowing a solution  $\mathcal{S}$  and given an expected performance level  $\mathcal{L}_\lambda$ , what is the neighbourhood  $\mathcal{P}$  of  $\mathcal{I}_{ref}$  such that  $\lambda(\mathcal{S}, z, \mathcal{P}) \leq \mathcal{L}_\lambda$ ?”.

This question can be seen as a stability analysis problem. Addressing this problem means finding the neighbourhood  $\mathcal{P}$  in which solution  $\mathcal{S}$  remains stable in the sense of the performance level  $\mathcal{L}_\lambda$ . This problem includes the first and second questions of sensitivity analysis defined in (Mahjoub et al., 2005):

- (1) What is the neighbourhood  $\mathcal{P}$  of  $\mathcal{I}_{ref}$  in which the solution  $\mathcal{S}_{\mathcal{I}_{ref}}^*$  remains optimal?  
Using the  $\mathcal{L}_\lambda$ -robustness formalism,  $\mathcal{S} = \mathcal{S}_{\mathcal{I}_{ref}}^*$ ,  $\lambda = \lambda_3$  and  $\mathcal{L}_\lambda = 0\%$ .
- (2) What is the neighbourhood  $\mathcal{P}$  of  $\mathcal{I}_{ref}$  in which the solution  $\mathcal{S}$  remains admissible with an acceptable performance?  
Now  $\mathcal{S}$  is supposed to be given,  $\lambda = \lambda_3$  and the value of  $\mathcal{L}_\lambda$  defines what is a so-called acceptable performance.

It must be stressed that these questions can also be stated using the stability radius introduced by (Sotskov et al., 1998).

*Definition 2.* In (Sotskov et al., 1998), given an instance  $\mathcal{I}$  and its optimal solution  $\mathcal{S}_{\mathcal{I}}^*$ , the stability radius  $\rho(\mathcal{S}_{\mathcal{I}}^*)$  is the maximum radius of a ball with centre  $\mathcal{I}$  in which the solution  $\mathcal{S}_{\mathcal{I}}^*$  remains optimal.

*Second robustness issue: sensitivity analysis.* It is assumed that in step 0, only a forecast instance  $\mathcal{I}_{ref}$ , a neighbourhood  $\mathcal{P}$  of  $\mathcal{I}_{ref}$  and a solution  $\mathcal{S}$  can be defined to characterise the optimisation problem. Then, the second robustness issue can be stated as follows: “knowing a solution  $\mathcal{S}$  and given a neighbourhood  $\mathcal{P}$  of  $\mathcal{I}_{ref}$ , what is the performance level  $\mathcal{L}_\lambda$  that is guaranteed by  $\mathcal{S}$  such that  $\lambda(\mathcal{S}, z, \mathcal{P}) \leq \mathcal{L}_\lambda$ ?”.

This question can be seen as a sensitivity analysis problem where the sensitivity is measured by  $\mathcal{L}_\lambda$  on the set of instances  $\mathcal{P}$ . This problem generalises the third question of sensitivity analysis defined in (Mahjoub et al., 2005):

3. Given  $\mathcal{I}$  a neighbour of  $\mathcal{I}_{ref}$ , is the solution  $\mathcal{S}_{\mathcal{I}_{ref}}^*$  still admissible and, if it is, what is the performance deviation?  
Using the  $\mathcal{L}_\lambda$ -robustness formalism,  $\mathcal{S} = \mathcal{S}_{\mathcal{I}_{ref}}^*$ ,  $\lambda = \lambda_2$  and  $\mathcal{P} = \{\mathcal{I}\}$ .

In our problem, the instance  $\mathcal{I}$  is generalised by the neighbourhood  $\mathcal{P}$ , and the performance deviation is assessed in the worst case on the neighbourhood.

*Third robustness issue: finding a robust solution.* It is assumed that in step 0, only a forecast instance  $\mathcal{I}_{ref}$ , a neighbourhood  $\mathcal{P}$  of  $\mathcal{I}_{ref}$  and a performance level  $\mathcal{L}_\lambda$  can be defined to characterise the optimisation problem. Then, the third robustness issue can be stated as follows: “knowing a performance level  $\mathcal{L}_\lambda$  that must be guaranteed on a given neighbourhood  $\mathcal{P}$  of  $\mathcal{I}_{ref}$ , what is a robust solution  $\mathcal{S}$  such that  $\lambda(\mathcal{S}, z, \mathcal{P}) \leq \mathcal{L}_\lambda$ ?”.

*Fourth robustness issue: maximising stability.* It is assumed that in step 0, only a forecast instance  $\mathcal{I}_{ref}$  and a performance level  $\mathcal{L}_\lambda$  can be defined to characterise the optimisation problem. Then, the fourth robustness issue can be stated as follows: “knowing a performance level  $\mathcal{L}_\lambda$  that must be guaranteed what is a solution  $\mathcal{S}$

that maximises the neighbourhood  $\mathcal{P}$  of  $\mathcal{I}_{ref}$  such that  $\lambda(\mathcal{S}, z, \mathcal{P}) \leq \mathcal{L}_\lambda$ ?”.

To answer this question, the neighbourhood  $\mathcal{P}$  covered by the solution  $\mathcal{S}$  must be measurable. For example, the neighbourhood  $\mathcal{P}$  can be measured by the stability radius (see definition 2). That means that the first issue must be addressed beforehand.

*Fifth robustness issue: minimising sensitivity.* It is assumed that in step 0, only a forecast instance  $\mathcal{I}_{ref}$  and a neighbourhood  $\mathcal{P}$  of  $\mathcal{I}_{ref}$  can be defined to characterise the optimisation problem. Then, the fifth robustness issue can be stated as follows: “giving a neighbourhood  $\mathcal{P}$  of  $\mathcal{I}_{ref}$  that must be covered, what is a solution  $\mathcal{S}$  that minimises the performance level  $\mathcal{L}_\lambda$  such that  $\lambda(\mathcal{S}, z, \mathcal{P}) \leq \mathcal{L}_\lambda$ ?”.

To answer this question the performance level  $\mathcal{L}_\lambda$  must be measurable. That means that the second robustness issue must be addressed beforehand.

## 5. CONFIGURATION OF A PARALLEL MULTI-PURPOSE MACHINES WORKSHOP WITH UNCERTAINTY

The aim of this section is to illustrate the previously presented robustness approach towards a manufacturing system application: a parallel multi-purpose machines workshop.

### 5.1 Configuration of a parallel multi-purpose machines workshop: step 0

Multi-purpose machines are used to model industrial workshops composed of parallel machines subject to constraints which prevent some machines from processing some product types (Brucker et al., 1997). These constraints may be technical and economical; they may be strict structural inability or be the result of a strategic choice. The configuration of such workshops is a key decision, the impact of which is significant on performance. The considered performance in this paper is the completion time (also referred to as the makespan). Thus, it is often observed that a human decision-maker designs or updates the workshop configuration manually, without having a quantitative measure of the configuration performance when unexpected demands have to be dealt with.

While initially designed to address the problem in the field of the semiconductor industry, the present work provides a theoretical framework that can be generalised to any workshop modelled as a set of multi-purpose machines (MPM) subject to demand uncertainty.

*A model for the MPM workshop and the attached scheduling problem.* In a multi-purpose machines workshop (Brucker et al., 1997),  $m$  machines and  $n$  types of products are considered, but all the machines cannot necessarily process all the product types: e.g., any machine can only process a subset of product types. The set of machines  $\{1, \dots, m\}$  is denoted  $M$ . The set of the product types  $\{1, \dots, n\}$  is denoted  $N$ .

The workshop machines are regarded as parallel machines and some *technological constraints* are used to prevent assigning products to a machine that cannot process them for technological reasons. All the technological constraints are modelled by a  $n$ -by- $m$  binary technological matrix  $T$ . Moreover, to be able to process one type of products, the machine must be qualified for the product type by undergoing a specific adjustment. The set of qualifications is modelled by a  $n$ -by- $m$  binary *configuration* matrix  $\mathcal{S}$ .  $\mathcal{S}$  is defined as follows:  $\mathcal{S}(i, j) = 1$  if machine  $j$  is qualified to process product type  $i$ , otherwise  $\mathcal{S}(i, j) = 0$ . A non-qualification may be the result of:

- (1) Technology: for some technological or physical reasons, machine  $j$  cannot process any product of type  $i$ . This information is given by  $T$  when  $T(i, j) = 0$  ( $T(i, j) = 0 \Rightarrow \mathcal{S}(i, j) = 0$ ).
- (2) Cost control: for economic reasons, all the machines may not be qualified for all the product types because this would be too expensive, and often useless (Aubry et al., 2008a).

To process the different product types, machines may have different speeds. A  $n$ -by- $m$  real *speed matrix*  $V$  is defined.  $V(i, j)$  is the number of products of type  $i$  that machine  $j$  is able to process during one unit of time.

The product types are the jobs to be processed by the machines. They are also referred to as the *demand*. The actual demand is modelled as a real  $n$ -column vector  $\mathcal{I}$ .  $\mathcal{I}(i)$  is the total amount of products of type  $i$  to be processed by the workshop for all  $i$  in  $N$ .

Scheduling the production in the workshop requires assigning the products to the machines. The result of the scheduling problem defines a *production plan* that can be modelled by a  $n$ -by- $m$  real matrix  $R$ .  $R(i, j)$  is the total amount of time that machine  $j$  spends processing products of type  $i$ . Furthermore, preemption and splitting are assumed: several machines may be working on the same product type at the same time. The order of the jobs on the machines does not matter, only the allocation matters. Given a demand  $\mathcal{I}$ , the speed data  $V$ , and the configuration  $\mathcal{S}$ , the scheduling problem consists in finding a production plan  $R$  such that the makespan  $C_{max}$  is minimised. Using the three fields notation introduced by (Graham et al., 1979), this problem can be stated as  $RMPM|split|C_{max}$ . *RMPM* stands for unrelated multi-purpose machines. *split* means that splitting is allowed.  $C_{max}$  means that the makespan (or the maximum completion-time of the machines) must be minimised. This problem is an easy problem (in the sense of the  $\mathcal{NP}$ -completeness theory) as it can be solved by the following linear program denoted  $LP$  which is inspired from the linear program of (Lawler and Labetoulle, 1978) used to solve  $R|pmtn|C_{max}$ :

$$LP \begin{cases} \min(C_{max}) \\ \sum_{j \in M} V(i, j) \times R(i, j) = \mathcal{I}(i) \quad \forall i \in N & (a) \\ \sum_{i \in N} R(i, j) \leq C_{max} \quad \forall j \in M & (b) \\ (1 - \mathcal{S}(i, j)) \times R(i, j) = 0 \quad \forall (i, j) \in N \times M & (c) \\ R(i, j) \geq 0 \quad \forall (i, j) \in N \times M & (d) \end{cases} \quad (2)$$

The speed  $V(i, j)$ , the configuration matrix  $\mathcal{S}(i, j)$  and the demand  $\mathcal{I}(i)$  are given data whereas  $R(i, j)$  and  $C_{max}$

are decision variables. The set of constraints (2a) enforces that the demand is exactly met for any type of products whereas the set of constraints (2b) enforces that  $C_{max}$  is greater than or equal to the completion-time of any machine in the shop. The set of constraints (2c) ensures that product type  $i$  cannot be assigned to machine  $j$  (e.g.,  $R(i, j) = 0$ ) if the machine is not qualified for this product type (e.g.,  $\mathcal{S}(i, j) = 0$ ). The last constraints define the variation domains of the variables  $R(i, j)$ .

In the following, we will denote the optimal value of the objective function of  $LP(V, \mathcal{S}, \mathcal{I})$  by  $C_{max}^{\mathcal{S}, V}(\mathcal{I})$ .

*Perturbations modelling.* It is first assumed that a forecast demand  $\mathcal{I}_{ref}$  is available. However, the actual demand (for the considered workshop) may be different because of unexpected events like production ratio deviation in upstream workshops, production losses and so on. The actual demand  $\mathcal{I}$  is regarded as the result of a variation  $\Delta \mathcal{I}$  around  $\mathcal{I}_{ref}$ :  $\mathcal{I} = \mathcal{I}_{ref} + \Delta \mathcal{I}$ . We define a neighbourhood  $\mathcal{P}_{a_1, \dots, a_n}(\mathcal{I}_{ref})$  around  $\mathcal{I}_{ref}$  by:

$$\mathcal{P}_{a_1, \dots, a_n}(\mathcal{I}_{ref}) = \left\{ \mathcal{I} \mid \mathcal{I} = \mathcal{I}_{ref} + \sum_{k \in N} \alpha_k \cdot a_k \cdot e_k \wedge \sum_{k \in N} \alpha_k \leq 1 \wedge \alpha_k \geq 0, \forall k \in N \right\} \quad (3)$$

with  $(e_k)_{k \in N}$  the canonical basis of  $\mathbb{R}_n^+$  and  $a_k$  the positive variation magnitude on the product type  $k$ .

It is assumed that the perturbations occur such that the actual instance belongs to a set of instances like  $\mathcal{P}_{a_1, \dots, a_n}(\mathcal{I}_{ref})$ .

*What must be guaranteed despite perturbations?* A deadline  $\tilde{d}$  is provided, assigning a maximum amount of time for completing any demand in the risk  $\mathcal{P}_{a_1, \dots, a_n}(\mathcal{I}_{ref})$  to be covered. In this context, the robustness of a configuration can be defined as follows.

*Definition 3.* Given the speed matrix  $V$  and the forecast demand  $\mathcal{I}_{ref}$ , a configuration  $\mathcal{S}$  is said to be robust on the set of demands  $\mathcal{P}_{a_1, \dots, a_n}(\mathcal{I}_{ref})$  if for all  $\mathcal{I}$  in  $\mathcal{P}_{a_1, \dots, a_n}(\mathcal{I}_{ref})$ ,  $C_{max}^{\mathcal{S}, V}(\mathcal{I}) \leq \tilde{d}$

The last definition means that the robustness criterion which can be used is the absolute robustness  $\lambda_1$ , after having set  $\mathcal{L}_\lambda = \tilde{d}$ . In other words,  $\mathcal{S}$  is said to be robust on  $\mathcal{P}_{a_1, \dots, a_n}(\mathcal{I}_{ref})$  if:

$$\lambda_1(\mathcal{S}, C_{max}^{\mathcal{S}, V}, \mathcal{P}_{a_1, \dots, a_n}(\mathcal{I}_{ref})) \leq \tilde{d}$$

## 5.2 Preliminary result

*Definition 4.* Let  $\mathcal{P}_{\tilde{d}}(\mathcal{S})$  be the set of demands that can be completed by the configuration  $\mathcal{S}$  in less than  $\tilde{d}$  units of time.

More formally,  $\mathcal{P}_{\tilde{d}}(\mathcal{S})$  can be defined as:

$$\mathcal{P}_{\tilde{d}}(\mathcal{S}) = \left\{ \mathcal{I} \mid C_{max}^{\mathcal{S}, V}(\mathcal{I}) \leq \tilde{d} \right\}$$

*Theorem 1.* (Aubry et al., 2008c)  $\mathcal{P}_{\tilde{d}}(\mathcal{S})$  is a convex set.

### 5.3 Stability analysis

In the addressed problem, the stability analysis problem consists in finding the largest set of demands  $\mathcal{P}_{a_1, \dots, a_n}(\mathcal{I}_{ref})$  that can be completed in less than  $\tilde{d}$  units of time by a given configuration  $\mathcal{S}$ . The resolution of this problem has been presented in (Aubry et al., 2008c). The resolution process relies on solving  $n$  linear programs, each one for determining the maximal value of  $a_k$ ,  $1 \leq k \leq n$ . Moreover, it is shown in (Aubry et al., 2008c) that the value defined by  $\min_{k \in N} a_k$  is the stability radius defined by (Sotskov et al., 1998).

### 5.4 Sensitivity analysis

For the considered example, the sensitivity analysis problem consists in finding the completion time  $\tilde{d}$  guaranteed by the configuration  $\mathcal{S}$  on the set of demands  $\mathcal{P}_{a_1, \dots, a_n}(\mathcal{I}_{ref})$ . In other words,  $\tilde{d}$  measures the completion time in the worst case on  $\mathcal{P}_{a_1, \dots, a_n}(\mathcal{I}_{ref})$ .

*Definition 5.* Let us denote  $\mathcal{I}_{ref}^{a_k}$  the demand such that:

$$\mathcal{I}_{ref}^{a_k} = \mathcal{I}_{ref} + a_k \cdot e_k$$

Note that each demand  $\mathcal{I}_{ref}^{a_k}$  belongs to  $\mathcal{P}_{a_1, \dots, a_n}(\mathcal{I}_{ref})$ .

*Theorem 2.*  $\tilde{d}$  is equal to  $\max_{k \in N} C_{max}^{S,V}(\mathcal{I}_{ref}^{a_k})$

*Proof.* Let set  $\tilde{d}$  to  $\max_{k \in I} C_{max}^{S,V}(\mathcal{I}_{ref}^{a_k})$ . By definition of  $\tilde{d}$ , each demand  $\mathcal{I}_{ref}^{a_k}$ ,  $1 \leq k \leq n$ , is such that the following inequality holds:

$$C_{max}^{S,V}(\mathcal{I}_{ref}^{a_k}) \leq \tilde{d} \quad (4)$$

That means that each demand  $\mathcal{I}_{ref}^{a_k}$  belongs to  $\mathcal{P}_{\tilde{d}}(\mathcal{S})$ .

Let  $\mathcal{I}$  be in  $\mathcal{P}_{a_1, \dots, a_n}(\mathcal{I}_{ref})$ . There is a family  $(\alpha_k)_{k \in N}$  of positive real numbers such that

$$\mathcal{I} = \mathcal{I}_{ref} + \sum_{k \in N} \alpha_k \cdot a_k \cdot e_k \text{ and } \sum_{k \in N} \alpha_k \leq 1.$$

Let  $\beta$  be the real positive number such that  $\beta = 1 - \sum_{k \in N} \alpha_k$ .

$\mathcal{I}$  can be written as follows:

$$\begin{aligned} \mathcal{I} &= \left( \beta + \sum_{k \in N} \alpha_k \right) \cdot \mathcal{I}_{ref} + \sum_{k \in N} \alpha_k \cdot a_k \cdot e_k \\ \Rightarrow \mathcal{I} &= \beta \cdot \mathcal{I}_{ref} + \sum_{k \in I} \alpha_k \cdot (\mathcal{I}_{ref} + a_k \cdot e_k) \\ \Rightarrow \mathcal{I} &= \beta \cdot \mathcal{I}_{ref} + \sum_{k \in I} \alpha_k \cdot \mathcal{I}_{ref}^{a_k} \end{aligned}$$

According to inequality (4) each demand  $\mathcal{I}_{ref}^{a_k}$  belongs to  $\mathcal{P}_{\tilde{d}}(\mathcal{S})$ . Moreover, by definition, the demand for each product type in  $\mathcal{I}_{ref}$  is less than the corresponding one in  $\mathcal{I}_{ref}^{a_k}$ ,  $1 \leq k \leq n$ , so the completion time to process  $\mathcal{I}_{ref}$  is less than  $\tilde{d}$ . That means that  $\mathcal{I}_{ref}$  belongs to  $\mathcal{P}_{\tilde{d}}(\mathcal{S})$  too.  $\mathcal{I}$  is thus a convex combination of elements belonging to  $\mathcal{P}_{\tilde{d}}(\mathcal{S})$ . According to the convex property of  $\mathcal{P}_{\tilde{d}}(\mathcal{S})$  (see theorem 1),  $\mathcal{I}$  belongs to  $\mathcal{P}_{\tilde{d}}(\mathcal{S})$  and so  $\mathcal{P}_{a_1, \dots, a_n}(\mathcal{I}_{ref}) \subseteq \mathcal{P}_{\tilde{d}}(\mathcal{S})$ . That means that each demand  $\mathcal{I}$  of  $\mathcal{P}_{a_1, \dots, a_n}(\mathcal{I}_{ref})$  is such that  $C_{max}^{S,V}(\mathcal{I}) \leq \tilde{d}$ . And by definition,  $\tilde{d}$  is reached for at

```
[Initialisation]
test, k : Integer
test ← 1
k ← 1
[Testing configuration S = T]
While (test = 1 AND k < n) do
    [Evaluating the makespan for demand I_ref^{a_k}]
    C_{max}^{S,V}(I_ref^{a_k}) ← LP(V, T, I_ref^{a_k})
    If (C_{max}^{S,V}(I_ref^{a_k}) ≤ d̃) then
        | k ← k + 1
    else
        | test ← 0
    end If
done
If (test = 1) then
    [Configuration S = T is robust]
    S ← T
else
    | There is no robust configuration
end If
```

Algorithm 1: Algorithm for finding a robust configuration

least one demand  $\mathcal{I}_{ref}^{a_k}$  of  $\mathcal{P}_{a_1, \dots, a_n}(\mathcal{I}_{ref})$ .  $\square$

### 5.5 Finding a robust configuration

In the context of multi-purpose machines, looking for a robust solution consists in finding a configuration  $\mathcal{S}$  such that each demand of  $\mathcal{P}_{a_1, \dots, a_n}(\mathcal{I}_{ref})$  can be completed in less than  $\tilde{d}$  units of time. If such a configuration exists, algorithm 1 yields one of them:  $\mathcal{S} = T$ . In fact the configuration having the largest slack time is the configuration  $\mathcal{S} = T$ : the machines are qualified for all the possible product types. This configuration is also the one that maximises the set of demands  $\mathcal{P}_{a_1, \dots, a_n}(\mathcal{I}_{ref})$  and minimises the deadline  $\tilde{d}$ . Thus if  $\mathcal{S} = T$  is not robust then there is no robust configuration.

Algorithm 1 uses the fact that  $\mathcal{P}_{\tilde{d}}(\mathcal{S})$  is convex for any  $\mathcal{S}$ . Thus only the extreme points of  $\mathcal{P}_{a_1, \dots, a_n}(\mathcal{I}_{ref})$  (i.e. the demands  $\mathcal{I}_{ref}^{a_k}$ ) have to be tested: if each demand  $\mathcal{I}_{ref}^{a_k}$  is completed in less than  $\tilde{d}$  units of time then all the demands in  $\mathcal{P}_{a_1, \dots, a_n}(\mathcal{I}_{ref})$  can be completed in less than  $\tilde{d}$  units of time.

If algorithm 1 does not return any robust configuration, that means that either  $\tilde{d}$  is too small or new efficient machines are needed. Identifying the characteristics of these machines is an interesting problem that remains open.

If configuration  $\mathcal{S} = T$  is an admissible configuration, it must be noted that it is also the most expensive one since it is the one having all the possible qualifications. That means that finding a trade-off between cost and robustness is a more relevant problem. This problem has been addressed in (Aubry, 2007). After having proved

that this problem is  $\mathcal{NP}$ -hard in the strong sense, the author proposes a Mixed-Integer Linear Program (MILP) to model it. Finally, the results found by solving this MILP with the free package GNU Linear Programming Kit (GLPK) are presented in (Aubry et al., 2008b).

### 5.6 Stability maximisation

For the considered example, maximising the stability consists in finding a configuration  $\mathcal{S}$  that covers the largest set of demands  $\mathcal{P}_{a_1, \dots, a_n}(\mathcal{I}_{ref})$  such that each one can be completed in less than  $\tilde{d}$  units of time.

The solution to this problem is easy to find. Once again, the configuration having the largest slack time is configuration  $\mathcal{S} = T$ . This configuration is also the one that maximises the set of demands that can be completed in less than  $\tilde{d}$  units of time. The largest set  $\mathcal{P}_{a_1, \dots, a_n}(\mathcal{I}_{ref})$  is thus defined by solving the stability analysis problem for  $\mathcal{S} = T$ .

As previously, it can be noted that configuration  $\mathcal{S} = T$  is also the most expensive one. That means that the trade-off between cost and stability is a more relevant problem. This problem remains open in the general case. However, performing an  $\epsilon$ -constraint approach (lower bounding  $\mathcal{P}_{a_1, \dots, a_n}(\mathcal{I}_{ref})$  and minimising the cost of the configuration) is solving the third problem.

### 5.7 Sensitivity minimisation

For the considered example, minimising sensitivity consists in finding a configuration  $\mathcal{S}$  that guarantees the shortest deadline  $\tilde{d}$  on the given set of demands  $\mathcal{P}_{a_1, \dots, a_n}(\mathcal{I}_{ref})$ .

As for the previous problem, the configuration having the largest slack time is  $\mathcal{S} = T$ . This configuration is also the one that minimises the deadline  $\tilde{d}$  on the given set of demands  $\mathcal{P}_{a_1, \dots, a_n}(\mathcal{I}_{ref})$ . The value for the minimal deadline  $\tilde{d}$  is thus returned by solving the sensitivity analysis problem for  $\mathcal{S} = T$ . That means that  $\tilde{d} = \max_{k \in N} C_{max}^{T,V}(\mathcal{I}_{ref}^{a_k})$ .

As previously, the trade-off between cost and sensitivity is a more relevant problem. This problem also remains open in the general case. However, performing an  $\epsilon$ -constraint approach (upper bounding  $\tilde{d}$  and minimising the cost of the configuration), is solving the third problem.

## 6. CONCLUSION

Taking uncertainty into account in optimisation problems has been dealt with in this paper. A generic off-line approach has been proposed and illustrated through an example. This example shows that the five robustness problems that are highlighted by the approach are relevant for industrial applications. This approach seems to be an interesting decision-aid tool for the production managers who want to face perturbations.

## REFERENCES

Aubry, A. (2007). *Optimisation pour la configuration robuste de systèmes de production de biens et de services*. Ph.D. thesis, Institut Polytechnique de Grenoble, Grenoble, France.

- Aubry, A., Rossi, A., Espinouse, M.L., and Jacomino, M. (2008a). Minimizing setup costs for parallel multi-purpose machines under load-balancing constraint. *European Journal of Operational Research*, 187(3), 1115–1125.
- Aubry, A., Rossi, A., and Jacomino, M. (2008b). On finding a minimum cost robust configuration for multipurpose machines. In *9ème congrès de la société Française de Recherche Opérationnelle et d'Aide à la Décision, ROADEF'08*, 29–30. Clermont-Ferrand, France.
- Aubry, A., Rossi, A., and Jacomino, M. (2008c). Sensitivity analysis for the configuration of a multi-purpose machines workshop. In *Proceedings of the 17th IFAC World Congress*, 15831–15836. Seoul, South Korea.
- Billaut, J.C., Moukrim, A., and Sanlaville, E. (2005). *Flexibilité et Robustesse en Ordonnancement*, chapter 1. Hermès Science, Paris, France.
- Brucker, P., Jurisch, B., and Krämer, A. (1997). Complexity of scheduling problems with multi-purpose machines. *Annals of Operational Research*, 70, 57–73.
- Graham, R., Lawler, E., Lenstra, J., and Kan, A.R. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, 5, 287–326.
- Kouvelis, P. and Yu, G. (1997). *Robust Discrete Optimization and its Applications*. Kluwer Academic Publishers, Dordrecht, the Netherlands.
- Lawler, E. and Labetoulle, J. (1978). On preemptive scheduling of unrelated parallel processors by linear programming. *Journal of the Association for Computing Machinery*, 25, 612–619.
- Mahjoub, A., Moukrim, A., and Sanlaville, E. (2005). *Flexibilité et Robustesse en Ordonnancement*, chapter 4. Hermès Science, Paris, France.
- Perny, P. and Spanjaard, O. (2006). A decision-theoretic approach to robust optimization in multivalued graph. *Annals of Operations Research*, 147, 317–341.
- Rossi, A. (2003). *Ordonnancement en milieu incertain, mise en œuvre d'une démarche robuste*. Ph.D. thesis, Institut National Polytechnique de Grenoble, France.
- Sotskov, Y., Wagelmans, A., and Werner, F. (1998). On the calculation of the stability radius of an optimal or an approximate schedule. *Annals of Operations Research*, 83, 213–252.
- Vincke, P. (1999). Robust solutions and methods in decision-aid. *Journal of Multi-Criteria Decision Analysis*, 8, 181–187.